

REMARKS

Claims 1-16 were pending and stand rejected. Claims 1 and 16 have been amended. New claims 17-20 have been added. All the amendments are fully supported by the specification.

Rejection Under 35 U.S.C. § 101

Claims 1-16 are rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. Applicants have amended claims 1-15 to recite “a computer-implemented method.” Claim 16 is amended to recite “a computer-implemented system.” Applicants respectfully submit that claims 1-16 as amended now recite statutory subject matter.

Rejection Under 35 U.S.C. 103 (a)

Claims 1-9 and 13-15 stand rejected under 35 U.S.C. § 103 (a) as being unpatentable over Shah and Minasi. Claims 10 and 11 stand rejected under 35 U.S.C. § 103 (a) as being unpatentable over Shah and Minasi and further in view of Shaw. Claim 12 stand rejected under 35 U.S.C. § 103 (a) as being unpatentable over Shah and Minasi and further in view of Debique. Claim 16 stands rejected under 35 U.S.C. § 103 (a) as being unpatentable over Shah in view of Minasi and further in view of Tanenbaum. Applicant respectfully traverses these rejections.

Independent claim 1, as amended, recites:

A computer-implemented method of efficiently provisioning application services for a plurality of diverse applications executed at a data center, the method comprising:
creating an organization entity within the data center;
creating an organization unit for the organization entity;
associating a group identification number with the organization entity;

creating user permission information for application services for the organization entity; and
propagating at least one of the organization unit, the group identification number, and the permission information to at least one application server within the data center. (Emphasis added).

In short, the claimed invention recites a method for provisioning application services for a plurality of diverse applications. The claimed invention creates an organization entity and an organization unit within a data center, associates a group identification number with the organization entity, creates user permission information for the organization entity, and propagates at least one organization unit, the group identification number, and the permission information to at least one application server within the data center. Propagating permission information is advantageous because it provides for centralized management of the permission information. In addition, when a user sends a request for application services, an application server does not need to query a database to determine whether a particular user has permission to access certain applications since permission information is propagated and thereby distributed to the application server.

Shah does not disclose or suggest the claimed invention. Although Shah discloses a mechanism that allows a user to access computer resources and remote applications, in Shah, user permission information is not propagated to at least one application server within the data center, as claimed. Indeed, in Shah, when a user defines a list of applications or resources he would like to have access, a business object is created on a fly that stores all information about the user. For example, at page 5, lines 4-10, Shah discloses:

When a user defines their toolset, a business object is built that is specifically suited to the user's needs. The business object can, for example, manifest itself as the Web page (and Java applets contained therein) through which the user gains access to the computing resources and applications unique to their task.

Similarly, at page 3, lines 20-30, Shah discloses:

Request processor 110 capable programmatically, in response to client-side requests from users, constructing dynamically extensible business objects that incorporate the required object services, and provide access to computing resources and applications unique to the user task at hand.

Thus, in Shah, all the information about a user is stored in a business object or a server object that is created specifically for that user. In fact, Shah does not even need a centralized management of user permission information because there is a one-to-one relationship between a business object and a user. Accordingly, when a user attempts to access application services, the user is connected to the business object specifically created for that user to access computing resources and applications that the user has selected (page 6, lines 10-30). Therefore, Shah does not disclose or suggest the claimed invention. Applicant respectfully traverses the Examiner's contention that the step of "propagating...permission information to at least one application server within the data center" is disclosed in Shah at page 5, lines 15-23, Fig. 1, and page 3, lines 26-31. Again, the cited portions of the reference simply disclose that all the user information is stored in business objects, and the business objects are stored in a business object database (page 5, line 21). Although Fig. 1 discloses an application server, there is no indication whatsoever that permissions are propagated at the application server level.

The addition of Minasi does not cure the deficiency of Shah. Minasi is an old article describing a conventional mechanism for managing the Windows NT Server 4 operating system. In Minasi, permissions and user accounts are stored in the Primary Domain Controller (PDC), an NT machine that holds a shared database of all users that have all agreed to constitute a domain (page 345). There is no disclosure or suggestion in Minasi, however, that permission information is propagated to at least one application server. Indeed, Minasi is concerned with providing access to network resources, rather than providing application services.

Further, the combination of Shah and Minasi would provide nothing beyond Shah itself, since Minasi makes no suggestion or recommendation as to how to extend or further implement Shah as to produce the claimed invention. Indeed, the combination of the two references would result in a system for provisioning application services to users, in which permission information is stored in a central database. In addition, it would not be obvious to modify Shah and Minasi to propagate user information including permissions to at least one application server. Indeed, Shah does not need to maintain centralized management of permission information since users access application services via business objects created for each individual user. Similarly, it is not obvious to modify Minasi to propagate user information to at least one application server because all Minasi is concerned with is providing access to network resources (e.g., a printer, a directory, a file, or some other network resource), rather than providing application services. Therefore, propagating user information to application servers would be meaningless in a situation described by Minasi.

Similarly, neither Shaw, Tanenbaum, or Debique cure the deficiencies of Shah and Minasi. Although Shaw discloses a mechanism for providing an application service for selection by a user, in Shaw user profiles and access information is saved in an administrative engine executed on the UAP server 250 (col. 7, lines 13-20). In Shaw, the system architecture is divided into three tiers—the first tier includes client devices, the second tier includes the UAP server 250, the web server, and other components, and the third tier includes various applications servers (col. 5, lines 30-35). The UAP server 250 is part of the second tier. There is no disclosure in Shaw that user profiles and access information are propagated to at least one application server.

Debique discloses a technique for verifying proper master replication of logical structures, such as objects, in a data processing system (col. 1, lines 30-35). In Debique,

replication is performed within a distributed data processing system having data processing resources, such as workstations, printers, secondary storage devices (col. 2, lines 63-67).

Although the reference discloses replication of logical structures from a first location to a second location in the data processing system, the system in Debique does not employ application servers. Thus, there is no disclosure in Debique that logical structures are replicated to at least one application server.

Tanenbaum discloses a redundant switching system. Again, Tanenbaum is not concerned with application service provisioning or propagating permission information to at least one application server. The cited portions of the reference do not disclose or suggest “propagating...permission information to at least one application server within the data center,” as claimed.

Thus, neither reference discloses or suggests “propagating at least one of the organization unit, the group identification number, and the permission information to at least one application server within the data center,” as claimed. The combination of the references provides nothing beyond Shah itself, since neither reference makes any suggestion or recommendation on how to further extend Shah as to produce the claimed invention.

Since neither reference discloses or suggests the claimed invention, either alone or in combination, claims 1 and 16 are patentable over the cited references. Claims 2-15 and 17-20 depend either directly or indirectly from independent claim 1 and 16 and derive their patentability from the independent claim from which they depend, in addition to reciting their patentable features.

New claims 17-20 are provided to specifically recite other novel features of the claimed invention.

For these reasons, Applicant respectfully submits that all the pending claims are allowable over the cited art of record and requests that the Examiner allow the case. In addition, Applicant respectfully requests that the Examiner change the old docket number 19134.0006 to the new docket number 23982-10381.

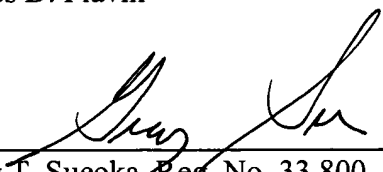
Respectfully submitted,

James D. Flavin

Dated: _____

9/15/05

By: _____



Greg T. Sueoka, Reg. No. 33,800

Fenwick & West LLP

Silicon Valley Center

801 California Street

Mountain View, CA 94041

Phone: (650) 335-7194

Fax: (650) 938-5200